2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Self-supervised Object Tracking with Cycle-consistent Siamese Networks

Weihao Yuan, Michael Yu Wang, and Qifeng Chen

***Abstract*— Self-supervised learning for visual object tracking possesses valuable advantages compared to supervised learning, such as the non-necessity of laborious human annotations and online training. In this work, we exploit an end-to-end Siamese network in a cycle-consistent self-supervised framework for object tracking. Self-supervision can be performed by taking advantage of the cycle consistency in the forward and backward tracking. To better leverage the end-to-end learning of deep networks, we propose to integrate a Siamese region proposal and mask regression network in our tracking framework so that a fast and more accurate tracker can be learned without the annotation of each frame. The experiments on the VOT dataset for visual object tracking and on the DAVIS dataset for video object segmentation propagation show that our method outperforms prior approaches on both tasks.**

I. INTRODUCTION

Visual object tracking is an essential task for numerous applications such as autonomous driving [1], robotic manipulation [2], and video surveillance [3]. Given the position of a target object in the first frame of a video, the task is to estimate its location in subsequent frames. In most cases, the tracking also needs to be run in real time. Although this problem has been studied by many researchers, state-of-the-art methods still suffer from many visual variations such as occlusion, deformation, motion, and illumination change [4, 5].

Recent deep network based methods [6–10] for visual object tracking have dominated most benchmarks and demonstrated advantages over traditional tracking methods [11–16] in both accuracy and speed. However, most deep-network-based methods require ground-truth object trajectories for training. The annotation of ground truth is laborious and time-consuming, which limits the size of the training data and their applications in unseen scenarios.

To solve this problem, some researchers are exploring self-supervised learning approaches [17] for object tracking. A tracker can be learned without the need for annotation on every frame. Additionally, self-supervised methods make online fine-tuning plausible such that they can be applied to unseen scenarios more easily. Nevertheless, previous self-supervised methods [17] are based on a correlation filter, and the performance is limited. On the other hand, Siamese network based trackers [7–10] have drawn much attention in the community recently. Siamese trackers formulate the visual object tracking task as learning a similarity response map by cross-correlation between the feature embedding of an exemplar patch and a search image. After the cross-correlation, a region proposal network or a mask regression network can contribute to the accurate prediction of the target.

Authors are with the Hong Kong University of Science and Technology, Hong Kong SAR, China. W. Yuan (weihao.yuan@connect.ust.hk) is with the Department of Electronic and Computer Engineering. M. Y. Wang is with the Department of Mechanical and Aerospace Engineering and the Department of Electronic and Computer Engineering. Q. Chen (cqf@ust.hk) is with the Department of Computer Science and Engineering and the Department of Electronic and Computer Engineering.

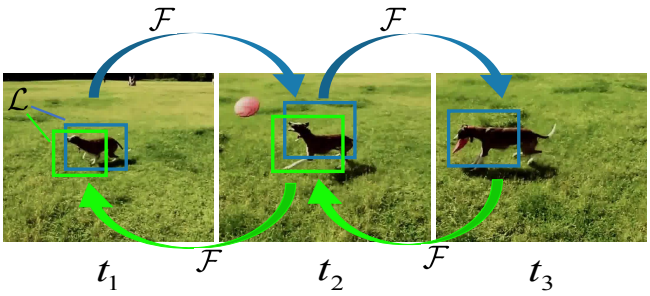

Fig. 1. Illustration of our self-supervised cycle-consistent framework. Given the bounding box (in blue) of the target object in the first frame, the object is tracked forward (in blue) in subsequent frames and then circularly tracked backward (in green) to the first frame. The discrepancy between the initial bounding box and the predicted box in the first frame can be used to serve as the supervision to optimize the tracking network $\mathcal{F}(\mathbf{z}, \mathbf{x}; \theta)$.

In this paper, we explore self-supervised learning for visual object tracking in a cycle-consistent fashion, utilizing the consistency between forward tracking and backward tracking, as shown in Fig. 1. For a video sequence, the tracking can be performed from the first frame to the last frame in chronological order, and then in reversed order back to the first frame. If the tracking is accurate, then the estimated location of the target object in the first frame after the backward tracking should be the same as the initial position. Thus, the discrepancy between these two positions can serve as the self-supervision to train the tracking network.

Following this idea, we introduce an end-to-end Siamese structure with the region proposal network [8, 18] into the cycle-consistent tracking framework. By leveraging the end-to-end learning ability of deep networks, our framework produces more accurate prediction than previous correlation-filter-based self-supervised methods [17]. The region proposal network (RPN) after the cross-correlation can estimate more accurate box proposals and improve the performance of the self-supervised tracking framework.

In addition, the cycle-consistent tracking can be also performed at the mask level, which is another fundamental task

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

in computer vision, the video object segmentation propagation (also called semi-supervised video object segmentation) [19–24]. Given the segmentation mask of the objects in the first frame in a video sequence, the task is to predict the segmentation in all the remaining frames. In our framework, we simply add a mask branch to predict the propagated mask, following the idea in [9]. This makes our framework for segmentation propagation still compact and can be run in real time, which is difficult for most approaches for this task, either supervised methods [19–22] or self-supervised methods [23, 24]. Also, the input of our network is a rough box of the target object rather than the accurate segmentation mask. These advantages mean that our method could be used in more practical applications.

In the experiments, we evaluate our framework on benchmark dataset VOT-2016 and VOT-2018 [5] for the visual object tracking task, and on DAVIS-2016 and DAVIS-2017 [4] for the segmentation propagation task. The results show accurate and stable performance of our cycle-consistent framework, with training on unlabeled video sequences. Taking advantage of the tracking target object initialization, our method outperforms previous state-of-the-art self-supervised approaches in both the visual object tracking task [17] and the video segmentation propagation task [23, 24], while running in real time.

Our contributions are then summarized as follows.

1) We introduce the Siamese region proposal network and mask regression module into the cycle-consistent framework to perform better end-to-end self-supervised learning.
2) The proposed method outperforms previous self-supervised algorithms in two tasks of visual object tracking and video segmentation propagation.

II. RELATED WORK

We first review the methods for visual object tracking in both supervised and self-supervised manners. Then, the approaches for supervised and self-supervised video segmentation propagation are surveyed.

Visual Object Tracking. In the past few years, correlation filter has shown to be fast and effective in comparing the difference between an exemplar image and its searching image due to its transformation in frequency domain, after proposed by Bolme *et al.* [11]. This filter has further been developed by introducing multi-channel [12], kernel [13], scale estimator [14], attention modular [15], and spatial relationship [16]. Recently deep-feature-based correlation filters [25, 26] have also been proposed for higher accuracy.

On the other hand, the Siamese structure deep trackers are growing rapidly and have dominated many benchmarks [6–10]. These Siamese network methods formulate visual object tracking as a cross-correlation problem between a template image patch and the searching image, and aim to exploit the mapping ability of deep networks from end-to-end learning. The template patch and the searching image are fed into Siamese networks, and the features are extracted in the same space, after which a cross-correlation is performed to merge two branches to one similarity response map. This structure has been demonstrated to be accurate and fast. To further improve the tracking accuracy, the region proposal network is applied to regress more accurate bounding boxes [8–10].

Deep trackers, however, rely heavily on the annotation labels for the training. To address this challenge, some researchers are exploring self-supervised methods for object tracking. As a widely-used unsupervised tool, auto-encoder is adopted to extract the generic image feature to detect the moving object in [27]. Another important constraint in visual tracking, the forward-backward error, is also widely used to estimate the error for optimizing the trackers [28], or labeling the annotations to provide more training data [29]. The forward-backward difference estimating is later extended with the geometry similarity, the cyclic weight, and the appearance similarity [30].

Recently the forward-backward checking idea has been applied to provide supervision in deep network training [17], which has improved the performance of self-supervised tracking. Nevertheless, there are still few deep trackers trained by self-supervision, and UDT [17] only uses a simple discriminative correlation filter to compare the features extracted from the exemplar and search region. Differently, to enhance the self-supervised cycle tracking framework, we use the depth-wise cross-correlation to generate a dense tensor and then feed it into the advanced region proposal network to regress the box location of the target object. In this way, end-to-end self-supervised learning is exploited to improve the cycle tracking framework.

Video Object Segmentation Propagation. Unlike visual object tracking, video segmentation propagation pays more attention to generating an accurate pixel-level mask for the target objects, such that the algorithms are usually time-consuming and are not in real time.

Traditional methods formulate segmentation propagation as a pixel matching problem. Optical flow is a widely used method to find pixel correspondence between two images but is not accurate for the object mask, so more mid-level features are introduced [31, 32]. Tsai *et al.* improve the optical flow by considering the object and spatial information [32], while SIFT flow considers the correspondence of SIFT features [31]. To find accurate pixel correspondence, graph labeling methods are also widely adopted, where a matching energy function is minimized for correspondence matching [19, 33, 34]. To better represent the pixels, deep features are extracted to compare the similarity between pixels [22].

Recent methods have tried to directly match the semantic correspondence according to deep features [35, 36]. Other methods try to process video frames independently [37, 38]. Not relying on temporal information, they fine-tune the trained network using the ground-truth mask provided in the first frame, making it totally a segmentation problem. Additionally, some methods try to propagate the initial mask from the first frame to subsequent frames once per frame, in a similar way to tracking [9].

Although the performance of deep learning methods is generally competitive, these approaches often require a large

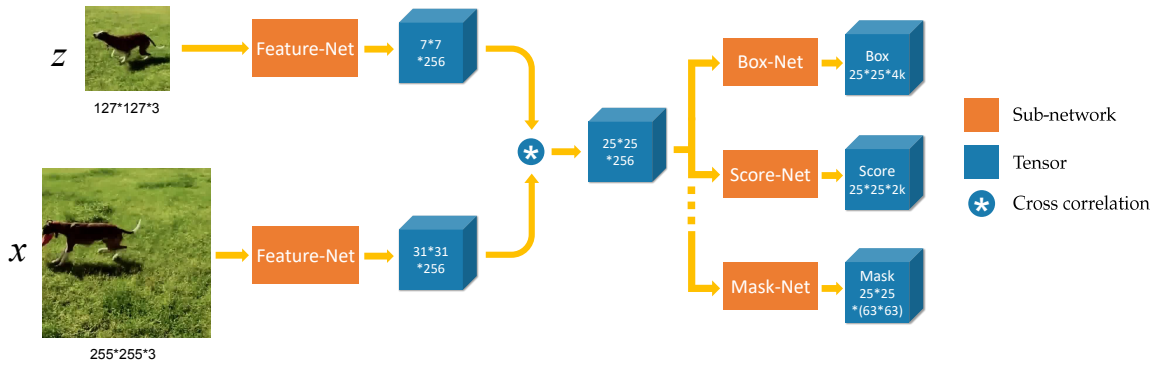

Fig. 2. The structure of our Siamese tracking network. The input includes two color images, which are the crop of the target object and the image where we need to search for the object. The output is a 25×25 dense response map, of which each element includes k box proposals and their corresponding scores. For mask tracking, each element also consists of a flattened mask with a size of $3969 = 63 \times 63$.

amount of data to do the training. Therefore, some self-supervised works have been proposed [23, 24, 39] based on cycle consistency. However, these methods are all first learning visual representations and then performing a correspondence matching. To better leverage the end-to-end learning of deep networks, we propose to add a mask branch in the cycle tracking framework, to learn a segmentation propagation function end-to-end with cycle consistency.

III. SELF-SUPERVISED TRACKING

In this section, we first introduce our cycle tracking framework, where the self-supervision works to optimize the tracking system. Then, the network structure and optimization function are given to illustrate how a single forward tracking in this framework is performed.

A. Cycle Object Tracking

Given a frame I_1 at time t_1, and the patch of the target object O to track, we first forward track the target O to frame I_2 at another time t_2. In the forward tracking, we can get the predicted location and size of the target object O in frame I_2:

$$\widetilde{p_2} = \mathcal{F}(p_1, I_2; \theta), \tag{1}$$

where $\mathcal{F}$ is the tracking network forwarding with parameters θ, p_1 is the patch of the target in frame I_1, and $\widetilde{p_2}$ is the predicted patch of the target in frame I_2.

As illustrated in Fig. 1, after the forward tracking, we backward track the patch $\widetilde{p_2}$ into the first frame I_1:

$$\widetilde{p_1} = \mathcal{F}(\widetilde{p_2}, I_1; \theta). \tag{2}$$

Then, we can get the predicted patch $\widetilde{p_1}$. Between $\widetilde{p_1}$ and p_1 we can calculate a consistency loss.

In this way, with the original patch as the label, and the predicted patch after cycle tracking as the output, we can obtain the loss to optimize the network parameters without the need for ground-truth annotations.

Furthermore, we can extend the tracking circle to more frames:

$$\begin{aligned} \widetilde{p_3} &= \mathcal{F}(\mathcal{F}(p_1, I_2), I_3; \theta), \\ \widetilde{p_1} &= \mathcal{F}(\mathcal{F}(\widetilde{p_3}, I_2), I_1; \theta). \end{aligned} \tag{3}$$

A longer circle makes the cycle tracking more challenging such that the network has to predict an accurate location of the target object in each single forward and backward tracking. The difference between more forward and backward pairs, such as p_2 and $\widetilde{p_2}$, could provide more supervision for the optimization.

B. Siamese Tracking Network

We follow [8, 9] to build a Siamese region proposal network, which has a template image patch $\mathbf{z}$ and a search image $\mathbf{x}$ as input. $\mathbf{z}$ is a small patch centered on the target object, and $\mathbf{x}$ is a large patch centered on the last predicted location of the object. These two patches are fed into two fully convolutional subnetworks sharing the same parameters to extract features, after which depth-wise cross-correlation and two branches, box-net, and score-net, are employed to produce a dense response map. The box-net generates multiple box candidates for each position in the response map. The score-net performs classification and outputs the object and background score for the corresponding box proposals. Therefore, each element in the response map comprises a set of k box proposals and their corresponding scores.

Each box proposal encodes 4 normalized coordinates following R-CNN [40]:

$$\begin{aligned} t_x = \frac{x - x_a}{w_a}, t_y = \frac{y - y_a}{h_a}, \\ t_w = \log\frac{w}{w_a}, t_h = \log\frac{h}{h_a}, \end{aligned} \tag{4}$$

where x, y, w, h denote the 2-dimensional coordinates of the center, width, and height of the predicted box, while x_a, y_a, w_a, h_a are for the anchor box.

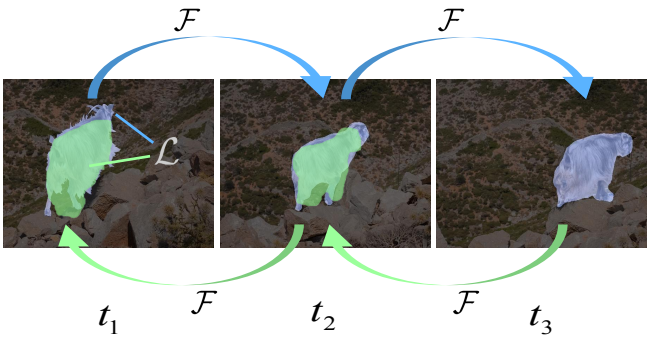

Fig. 3. Illustration of the self-supervised cycle segmentation propagation framework. Given the blue mask of the target object in the frame, the mask is forward propagated to subsequent frames and then circularly propagated back to the initial frame, generating the predicted mask in green. The forward propagation is indicated by blue arrows while the backward propagation is denoted by green.

The score of each box is composed of positive and negative activation $s_{\text{obj}}, s_{\text{back}}$, which are then processed by the softmax function to encode the probability of the box representing an object p_{obj} and the background p_{back}.

Thus, the output of the region proposal subnetwork is a $4k$ channel box vector and a $2k$ channel score vector, as displayed in Fig. 2.

C. Loss Function

During training, we do not calculate the loss of tracking in the middle of the circle. After a whole circle of the forward and backward tracking, we calculate the loss between the prediction and the initial target.

The box localization loss for each box is calculated with smooth L_1 loss and is formulated as

$$\mathcal{L}_{\text{box}} = l_1(t_x - t_x^*) + l_1(t_y - t_y^*) + l_1(t_w - t_w^*) + l_1(t_h - t_h^*), \tag{5}$$

where

$$l_1(x) = \begin{cases} \frac{1}{2}x^2 & |x| < 1 \\ |x| - \frac{1}{2} & otherwise \end{cases},$$

and t^* is the target box label.

The object score loss for each box is calculated by cross-entropy loss and formulated as

$$\begin{aligned} \mathcal{L}_{\text{sco}} = -[&y_{\text{o}} \log(p_{\text{obj}}) + (1 - y_{\text{o}}) \log(1 - p_{\text{obj}}) + \\ &y_{\text{b}} \log(p_{\text{back}}) + (1 - y_{\text{b}}) \log(1 - p_{\text{back}})] \end{aligned}, \tag{6}$$

where y_{o} and y_{b} are the object label and the background label.

Then, the final loss is a weighted sum of these two branches, as

$$\mathcal{L} = \mathcal{L}_{\text{sco}} + \lambda_1 \mathcal{L}_{\text{box}}, \tag{7}$$

where the λ_1 is a weighting factor. With this loss, the cycle tracking framework can be optimized by the self-supervision without the need for expensive annotations.

IV. Self-Supervised Segmentation Propagation

Our cycle-consistent tracking can be extended beyond bounding box tracking. In this section, we first extend the idea for visual object tracking to video segmentation propagation in a self-supervised manner, after which the mask prediction branch is added to assist the segmentation tracking task.

A. Cycle Mask Propagation

With a similar idea in cycle object tracking, we can address the tracking problem at the mask level, i.e., video object segmentation propagation. Given a frame I_1 at time t_1, and the mask of the target object to track, we can first forward propagate the mask to frame I_3 at time t_3, and then backward propagate the mask to frame I_1 circularly. Then, between the initial mask and the predicted mask in frame I_1, we can obtain a consistency loss as the supervision of this cycle propagation flow, as illustrated in Fig. 3. Therefore, the network can be trained without the need for annotations of every frame.

B. Siamese Mask Network

To facilitate the mask propagation in practical applications, following the idea of [9], the input of the mask propagation network is the same as the box tracking network, i.e., a template image patch $\mathbf{z}$ and a search image $\mathbf{x}$ as input. After the depth-wise cross-correlation, now there are three branches for this network: a score-subnetwork, a box-subnetwork, and an additional mask-subnetwork, as shown in Fig. 2. With the tensor after cross-correlation as input, for each position of the response map, the mask-net outputs a flattened vector of size $w^m \times h^m$, representing a mask prediction with width w^m and height h^m. This mask is resized to the shape of the original search image in inference.

After one propagation circle, the mask loss is calculated for each mask candidate in the response map [9] by

$$\mathcal{L}_{\text{mask}} = \sum_n \left(\frac{1 + y_n}{2w^m h^m} \sum_{i,j} \log(1 + e^{-c_n^{ij} m_n^{ij}})\right), \tag{8}$$

where $m_n^{ij} \in \{\pm 1\}$ is the predicted mask label for pixel (i, j) of n-th mask candidate, c_n^{ij} denotes the label of the target, and $y_n \in \{\pm 1\}$ denotes if the element in the response map is positive. The element is considered positive ($y_n = 1$) if one of its k boxes has more than 0.6 IoU with the target box.

Then, in the mask propagation network, the final loss consists of three terms:

$$\mathcal{L} = \mathcal{L}_{\text{sco}} + \lambda_1 \mathcal{L}_{\text{box}} + \lambda_2 \mathcal{L}_{\text{mask}}, \tag{9}$$

where λ_1 and λ_2 are two weighting factors.

V. Experiments

In this section, we first describe in details how we implement the cycle tracking framework and then present the quantitative and qualitative experiments to evaluate our method on both the visual object tracking task and the video object segmentation propagation task.

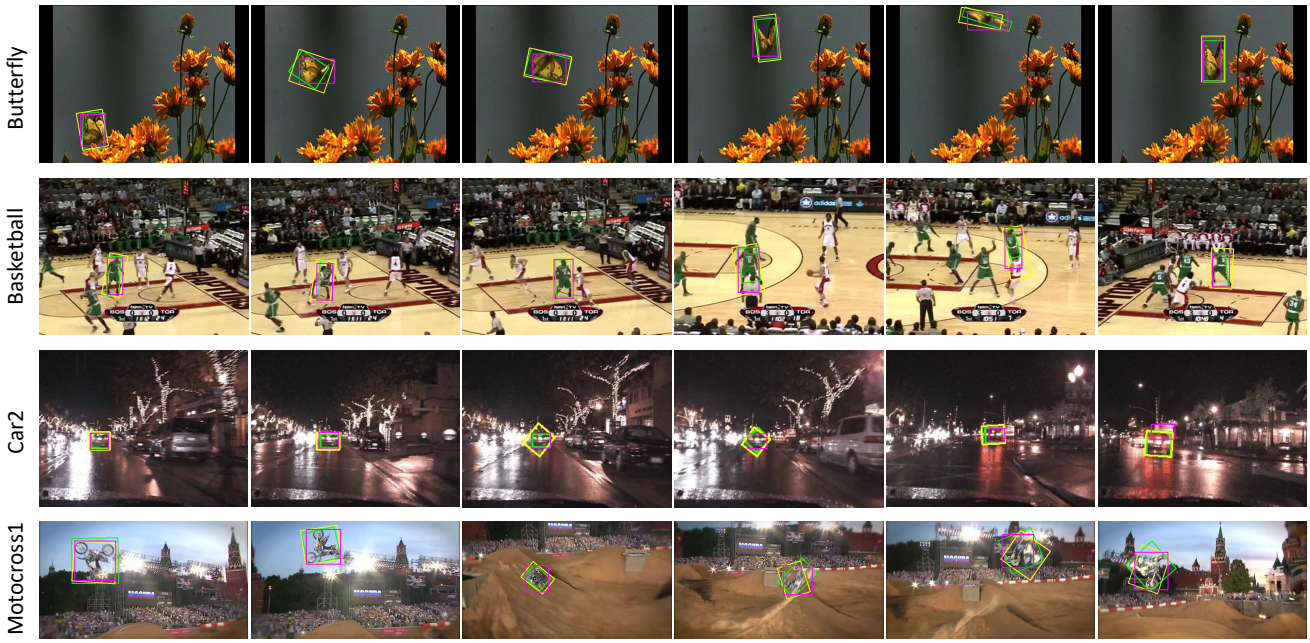

Fig. 4. Qualitative results of visual object tracking on the VOT dataset. The ground-truth bounding box in green illustrates the target object to track. The predictions of CycleSiam and CycleSiam$^+$ are denoted by pink and yellow boxes, respectively.

A. Implementation details

The self-supervised cycle tracking framework is implemented in Pytorch and trained on one Nvidia GTX 1080 Ti GPU. The batch size is set to 32, and the weights λ_1 and λ_2 are set to 1 and 30 without careful searching. The network is optimized end-to-end with SGD, where the learning rate is 0.001. The input image size for template $\mathbf{x}$ and search $\mathbf{z}$ is 255×255 and 127×127, respectively.

In the training, the anchor number k is set to 5 with ratios $[0.33, 0.5, 1, 2, 3]$ and scale 8. The object label is defined at where the anchors have $IoU > 0.6$ with the corresponding ground-truth box, and the background label is defined at where the anchors have $IoU < 0.3$ with the ground-truth. The score losses for other anchors are ignored.

We only calculate the loss and optimize the network parameters after one whole cycle of the tracking is finished. In the middle of the cycle tracking, we do not calculate the loss and simply regard it as an inference. For inference, we crop the last predicted box as the template patch $\mathbf{z}$, and crop the image centered on the last prediction as the search patch $\mathbf{x}$. The output box and mask are selected according to the maximum score in the score classification branch. For the mask branch, after a per-pixel sigmoid is applied, we binarize the mask with the threshold of 0.5 to get the final mask output.

B. Self-supervised Visual Object Tracking

Training. For a fair comparison with existing tracking methods, we train our model on the ILSVRC-2015 dataset. The object box to be tracked is set to be the initial target of the cycle tracking framework, then forward and backward tracking is performed to get the predicted bounding box in the first frame. Between the prediction and initial target object, we calculate the loss and optimize the network.

Method	Accuracy	Robustness	EAO	Speed (fps)
SCT [15]	0.462	0.545	0.188	40
DSST [14]	0.533	0.704	0.181	25
KCF [13]	0.489	0.569	0.192	170
UDT [17]	0.54	0.475	0.226	70
UDT+ [17]	0.53	0.308	0.301	55
CycleSiam	**0.603**	0.294	0.371	59
CycleSiam$^+$	0.601	**0.247**	**0.398**	44

TABLE I. Quantitative results of visual object tracking on VOT-2016. Accuracy, robustness, EAO, and speed are reported.

Method	Accuracy	Robustness	EAO	Speed (fps)
DSST [14]	0.395	1.452	0.079	25
KCF [13]	0.447	0.773	0.135	170
HMMTxD [41]	0.506	0.815	0.168	–
CycleSiam	**0.562**	0.389	0.294	59
CycleSiam$^+$	0.549	**0.314**	**0.317**	44

TABLE II. Quantitative results of visual object tracking on VOT-2018.

Evaluation. After the training, we evaluate our self-supervised object tracking framework on the VOT-2016 and VOT-2018 datasets [5] without fine-tuning. Both datasets include 60 video sequences, and there are hundreds of frames in each sequence. Some example sequences are shown in Fig. 4. Beginning with the first frame, we run the tracking network once per frame according to the official policy.

The quantitative results are summarized in TABLE I and

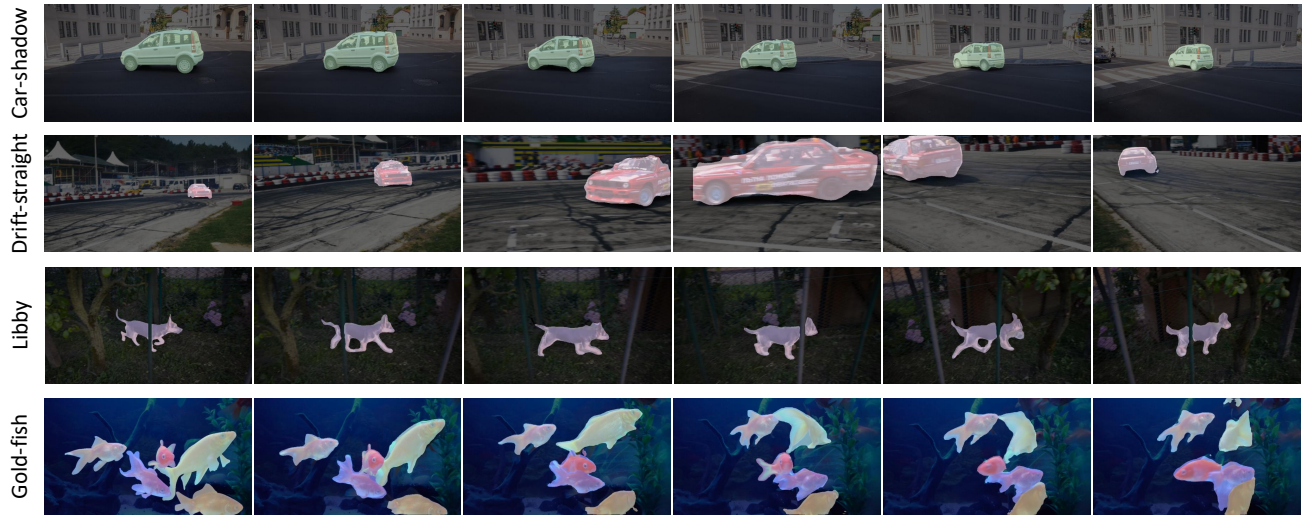

Fig. 5. Qualitative results of instance mask propagation on the DAVIS-2017 dataset. The first image of every sequence is the input with ground-truth annotation. Different instances are denoted with different colors. Given only the coarse box of the object, our network can predict its instance mask in the subsequent frame in real time.

Initialization	Dataset	Accuracy	Robustness	EAO
Random	VOT-2016	0.540	0.735	0.191
	VOT-2018	0.377	0.750	0.131
Object	VOT-2016	0.603	0.294	0.371
	VOT-2018	0.562	0.389	0.294

TABLE III. Ablation study with random target initialization.

TABLE II. We report the running speed and three official metrics, accuracy, robustness, and expected average overlap (EAO). Since there are few self-supervised deep methods for visual object tracking, we also include some traditional trackers. From the results, we can see that our CycleSiam reaches the EAO of 0.371 in the VOT-2016 dataset and outperforms the previous state-of-the-art self-supervised method UDT by a large margin.

The qualitative results are displayed in Fig. 4. Box predictions of different setup are denoted by different colors. Our method can track the object stably even when the target is mixed with other objects, e.g., the second and fourth images of the Basketball sequence, or when the video is quite blurry, e.g., the last three images of the Motocross sequence. Sometimes our tracker can give the prediction that is more reasonable than the ground-truth label, e.g., the fourth and fifth image of the Basketball sequence, and the first image of the Motocross sequence.

Ablation study. In addition, to test if our framework can work in arbitrary video sequences, we perform an ablation study with random target initialization. The initial target box is no longer given by the tracking target object input, but a random box in the image. In this case, the randomly set target box may contain multiple objects, meaningless background, or parts of an object. This means sometimes the network can be confused by these training data. The performance of our model trained on these messy data is reported in TABLE. III.

Method	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)	Speed (fps)
FCP [33]	58.4	49.2	–
BVS [34]	60.0	58.8	3
CycleSiam$^+$	**64.9**	**62.0**	31

TABLE IV. Quantitative results of video object segmentation propagation on DAVIS-2016. $\mathcal{J}$ is the Jaccard index and $\mathcal{F}$ is the contour F-measure.

Method	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)	Speed (fps)
SIFT Flow [31]	33.0	35.0	–
Transitive-Inv [42]	32.0	26.8	–
DeepCluster [43]	37.5	33.2	–
Wang et al. [23]	41.9	39.4	–
Lai et al. [24]	48.4	52.2	–
CycleSiam$^+$	**50.9**	**56.8**	31

TABLE V. Quantitative results of video object segmentation propagation on DAVIS-2017.

Although the performance is lower, it can still work well in most videos. But the model is confused in some difficult frames where the target object is mixed with the background or other distraction objects. Also, the accuracy is limited, e.g., the predicted box cannot fit tightly to the ground-truth.

C. Self-supervised Video Object Segmentation Propagation

Training. For the video segmentation propagation task, we use the network with three branches. The propagation network is trained on the YouTube-VOS dataset [44]. The mask of the target object to be propagated in the first frame is set to be the target mask initialization. Then, the axis-aligned bounding box is extracted from the mask, as the input of our network. Afterward, the forward and backward propagation is performed to get the predicted mask in the first frame. Between the prediction and the original mask,

we calculate the loss and optimize the network.

Evaluation. After the training, we evaluate our model on the video object segmentation task on the DAVIS-2016 and DAVIS-2017 [4] validation set without fine-tuning. Given the initial instance mask of the first frame, we extract the axis-aligned bounding box and track the mask in subsequent frames in turn. Similar to most video object segmentation approaches, for multiple instance cases, multiple inferences are performed at the same time.

The performances of our method, denoted by CycleSiam$^+$, and of other self-supervised methods are presented in TABLE V. The two official metrics, Jaccard index $\mathcal{J}$ for region similarity and contour F-measure $\mathcal{F}$ for contour similarity, are reported. Since there are few self-supervised methods for video segmentation propagation, we also include some visual feature works and use them to find segmentation correspondence, such as SIFT flow [31], Transitive-Inv [42], and DeepCluster [43], of which the performance is calculated in [23]. Additionally, the performance of some traditional methods like FCP [33] and BVS [34] are reported.

From the results, we can see that our method outperforms all previous self-supervised algorithms. Please note that the state-of-the-art methods usually use multiple previous frames (7 frames in [23]) as the input to predict the mask in the next frame, while we only use the current single frame. In addition, we report our online speed. One important advantage of our method is that our method can be run in real time. Also, our method only requires a simple bounding box as input. The qualitative results for some videos are presented in Fig. 5. From the visual results, our method can propagate the instance mask stably even when the object size varies significantly, like the Drift-straight sequence. But the mask prediction is not accurate enough in some videos since our method is trained in a self-supervised manner.

Now that we can track the object at the mask level, we can generate a rotated bounding box from the mask. The minimum bounding rectangle for the mask is generated as the box prediction. We evaluate the boxes generated from masks on the VOT-2016 and VOT-2018 datasets and obtain better performance, as reported in TABLE I and TABLE II. From the visualization in Fig. 4, the rotated box can fit the ground-truth better since the annotation in the VOT dataset is also a rotated box.

VI. CONCLUSION

In this work, we exploit the end-to-end Siamese network in cycle tracking to perform better self-supervised learning for visual object tracking and video object segmentation propagation. By taking advantage of the cycle consistency in a forward and backward tracking circle, self-supervision can be obtained. For the visual object tracking task, the target object is first forward tracked to subsequent frames and then traced back to the first frame. The loss is obtained from the difference between the initial bounding box and the estimated bounding box in the first frame. For the video object segmentation propagation task, in a similar way, the mask is first forward propagated and then circularly propagated back to the first frame, where the consistency loss is calculated to optimize the network.

To leverage the end-to-end learning of deep networks, we introduce the Siamese region proposal network and mask regression network into the tracker, such that a fast and more accurate tracker can be trained end-to-end. In the evaluation experiments on visual object tracking and video object segmentation propagation benchmark datasets, our method outperforms state-of-the-art self-supervised methods in both tasks. In the visual object tracking task, we outperform previous methods by a large margin. In the video segmentation propagation task, we need only a rough bounding box of the objects in the current frame to infer the mask in the next frame, while other methods often use multiple preceding frames. Additionally, our method can be run in real time. These advantages mean that our method could be useful in more practical applications.

VII. ACKNOWLEDGEMENT

This work is supported by the Innovation and Technology Fund of the Government of the Hong Kong Special Administrative Region (Project No. ITS/018/17FP, ITS/104/19FP).

REFERENCES

[1] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 3029–3037.

[2] C. Choi and H. I. Christensen, "Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4048–4055.

[3] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[4] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[5] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey *et al.*, "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[6] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1420–1429.

[7] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 850–865.

[8] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[9] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[10] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, 2019, pp. 4282–4291.

[11] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2544–2550.

[12] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proceedings of the International Conference on Computer Vision*, 2013, pp. 3072–3079.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[14] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[15] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4321–4330.

[16] A. Lukezic, T. Vojir, L. Cehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6309–6318.

[17] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1308–1317.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[19] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang, "Jots: Joint online tracking and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2226–2234.

[20] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2663–2672.

[21] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the International Conference on Computer Vision*, 2017, pp. 686–695.

[22] L. Bao, B. Wu, and W. Liu, "Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5977–5986.

[23] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2566–2576.

[24] Z. Lai and W. Xie, "Self-supervised learning for video correspondence flow," in *Proceedings of the British Machine Vision Conference*, 2019, p. 299.

[25] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6638–6646.

[26] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2805–2813.

[27] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, 2013, pp. 809–817.

[28] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[29] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 300–317.

[30] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Multihypothesis trajectory analysis for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5088–5096.

[31] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2010.

[32] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3899–3908.

[33] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung, "Fully connected object proposals for video segmentation," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 3227–3234.

[34] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, "Bilateral space video segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 743–751.

[35] N. Ufer and B. Ommer, "Deep semantic feature matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6914–6923.

[36] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6917–6925.

[37] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 221–230.

[38] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," in *Proceedings of the British Machine Vision Conference*, 2017.

[39] X. Li, S. Liu, S. De Mello, X. Wang, J. Kautz, and M.-H. Yang, "Joint-task self-supervised learning for temporal correspondence," in *Advances in Neural Information Processing Systems*, 2019, pp. 318–328.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[41] T. Vojir, J. Matas, and J. Noskova, "Online adaptive hidden markov model for multi-tracker fusion," *Computer Vision and Image Understanding*, vol. 153, pp. 109–119, 2016.

[42] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *Proceedings of the International Conference on Computer Vision*, 2017, pp. 1329–1338.

[43] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[44] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018.